บทที่ 2
หลักการและทฤษฎี

2.1 พื้นฐานทั่วไปของสเต็ปเปอร์มอเตอร์ [6]

สเต็ปเปอร์มอเตอร์เป็นมอเตอร์อีกชนิดหนึ่งที่ถูกนำมาใช้งานกันมาก ไม่ว่าจะเป็นส่วนประกอบในหุ่นยนต์ เครื่องจักรกลอัตโนมัติควบคุมด้วยคอมพิวเตอร์ ต่างก็ต้องใช้สเต็ปเปอร์มอเตอร์เป็นตัวขับเคลื่อน ตัวอย่างใกล้ตัวที่มีให้เห็น เช่น ฟลอบปี้ไดรฟ์ ฮาร์ดดิสก์ไดรฟ์ภายในคอมพิวเตอร์ ซึ่งก็ใช้สเต็ปเปอร์มอเตอร์ขนาดเล็กในการขับเคลื่อนหัวอ่านเขียนข้อมูลลงไปในพื้นผิวของแผ่นดิสก์เพื่อให้ตรงกับตำแหน่งของแทร็กที่ถูกต้อง

2.1.1 ชนิดของสเต็ปเปอร์มอเตอร์

การแบ่งชนิดของสเต็ปเปอร์มอเตอร์หากแบ่งตามโครงสร้างพื้นฐานหรือความแตกต่างของรูปแบบโรเตอร์จะแบ่งออกได้ 4 ชนิด แต่ถ้าแบ่งตามวิธีการพันขดลวดบนสเตเตอร์จะแบ่งออกได้ 2 ชนิด

1. แบ่งตามโครงสร้างพื้นฐาน

- *ชนิดวาริเอเบิลรีลักแตนซ์ (Variable Reluctance ; VR)*

มีโครงสร้างโรเตอร์แบบมัลติทูธ (multi-tooth) ทำจากเหล็กอ่อน จะทราบว่าเป็นมอเตอร์ชนิดนี้โดยการทดสอบได้ง่ายมาก คือใช้มือหมุนเพลามอเตอร์ และสังเกตเห็นว่าหมุนได้ตลอดโดยไม่ติดขัด เพราะโรเตอร์จะไม่เกิดปรากฏการณ์ทางแม่เหล็กแตกต่างจากชนิด PM และชนิดไฮบริด ซึ่งมีสนามแม่เหล็กที่โรเตอร์ ขณะหมุนจะรู้สึกขัดๆเหมือนเป็นฟันเฟือง ชนิดนี้มีจุดด้อยในความถูกต้องของตำแหน่งและทำงานได้ไม่ดีนักเมื่อมีสเต็ปการหมุนสูง

- *ชนิดแม่เหล็กถาวร (Permanent Magnet ; PM)*

มีโครงสร้างของโรเตอร์แบบเรียบไม่มีซี่ขั้วแม่เหล็ก บนโรเตอร์จะเป็นแบบแม่เหล็กถาวร การควบคุมทำได้โดยป้อนกระแสกระตุ้นที่ขดลวดบนสเตเตอร์ เช่นถ้าเป็นสเตเตอร์แบบ 4 เฟส จะมีขั้วแม่เหล็กอยู่ 4 ขั้ว ซึ่งมีคอยล์พันแยกจากกัน ขั้วแม่เหล็กถาวรบนโรเตอร์จะถูกแรงดึงดูดจากขั้วแม่เหล็กบนสเตเตอร์เมื่อป้อนกระแสไฟฟ้าเข้าสู่ขดลวด โรเตอร์จะอยู่คงที่ที่ขั้วแม่เหล็กบนสเตเตอร์นั้นถึงแม้ว่าจะไม่ป้อนกระแสไฟฟ้าอีกต่อไป ทำให้เกิดเป็นแรงยึดหน่วงขึ้น ชนิดนี้มีข้อดีในความถูกต้องของตำแหน่ง แม้ความเร็วจะมากขึ้นเมื่อเปรียบเทียบกับชนิดอื่น

- *ชนิดไฮบริด (Hybrid)*

เป็นชนิดที่นิยมใช้งานกันมากที่สุด โดยเฉพาะนำมาใช้กับอุปกรณ์ในเครื่องคอมพิวเตอร์ โครงสร้างได้จากการรวมเอาโครงสร้างของโรเตอร์ชนิดวาริเอเบิลรีลักแตนซ์และชนิดแม่เหล็กถาวรมาประกอบเข้าด้วยกัน จึงทำให้เป็นมอเตอร์ชนิดที่มีแรงยึดหน่วงสูง มีแรงบิดดึงและผลักได้ดีซึ่งมีความคงที่และทำงานได้ดีถึงแม้ว่าจะมีสเต็ปต่อรอบในการหมุนสูง

- ชนิดแรเอิร์ธเพอร์มาเนนต์แมกเน็ต

เป็นสเต็ปเปอร์มอเตอร์แบบใหม่อีกชนิดหนึ่งปรับปรุงมาจากชนิดชนิดแม่เหล็กถาวร มีโครงสร้างของโรเตอร์เป็นแผ่นยึดติดกับเพลามอเตอร์ มีโมเมนต์ความเฉื่อยต่ำ อัตราเร่งสูง แรงบิดดึงกำลังทางกลและความถูกต้องของตำแหน่งสูงมาก ความเร็วเริ่มหมุนและหยุดสูง สูญเสียพลังงานต่ำ ชนิดนี้มีชื่อเรียกอีกอย่างว่า ดิสก์แมกเน็ตสเต็ปเปอร์มอเตอร์ (Disc magnet steppers)

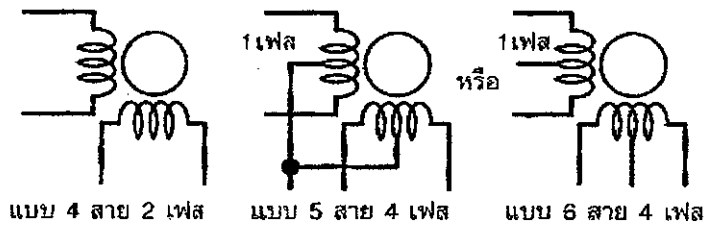
2. แบ่งตามการพันขดลวดบนสเตเตอร์

- แบบไบโพลาร์ (Bipolar)

มีการพันขดลวด 1 ขดบนแต่ละขั้วแม่เหล็กของสเตเตอร์ ขั้วแม่เหล็กที่เกิดขึ้นบนสเตเตอร์ถูกกำหนดโดยทิศทางของการะแสไฟฟ้า และทำให้เกิดขั้วแม่เหล็กในทิศทางการไหลและการกลับทิศการไหลของกระแสไฟฟ้า ทำได้โดยการใช้วงจรสวิตชิ่งกลับขั้วไฟฟ้า

- แบบยูนิโพลาร์ (Unipolar)

มีการพันขดลวด 2 ขดบนแต่ละขั้วแม่เหล็กของสเตเตอร์ ซึ่งแต่ละขดจะทำให้เกิดขั้วแม่เหล็กในทิศทางตรงข้ามกัน การพิจารณาว่าสเต็ปเปอร์มอเตอร์ตัวใดมีการพันขดลวดแบบใดสังเกตได้ง่าย ถ้าเป็นแบบไบโพลาร์จะมีสายไฟต่อออกจากมอเตอร์เพียง 4 สาย และถ้าเป็นแบบยูนิโพลาร์จะมี 5-6 สาย หรือสังเกตได้จากป้ายชื่อที่ติดอยู่กับมอเตอร์



(ก) แบบไบโพลาร์ (ข) แบบยูนิโพลาร์

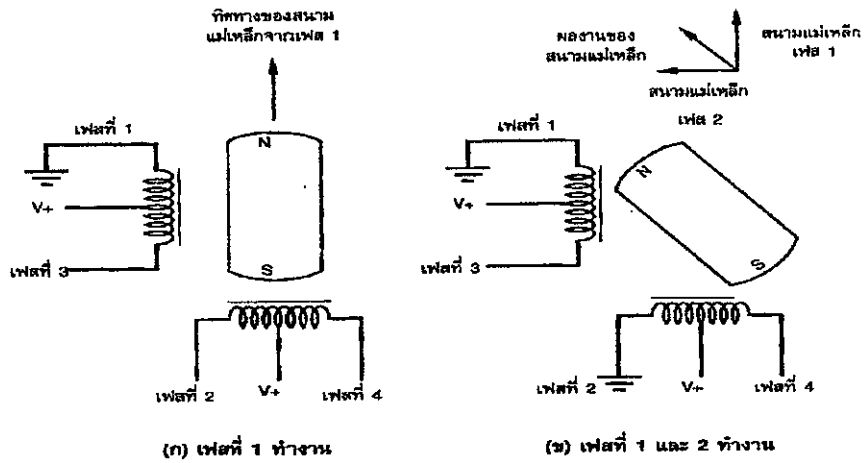
รูปที่ 2.1 ลักษณะการพันขดลวดบนสเตเตอร์

2.1.2 การทำงานของสเต็ปเปอร์มอเตอร์

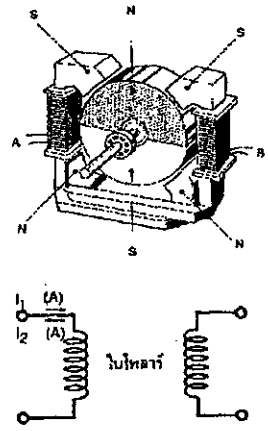
การทำงานของสเต็ปเปอร์มอเตอร์มีความแตกต่างจากมอเตอร์ทั่ว ๆ ไปโดยเมื่อป้อนกำลังไฟฟ้าให้มอเตอร์ชนิดนี้ก็จะหมุนเพียงเล็กน้อยตามเส้นรอบวงและหยุด แต่มอเตอร์ทั่วไปจะหมุนทันทีและตลอดเวลา สเต็ปเปอร์มอเตอร์สามารถกำหนดตำแหน่งการหมุนด้วยตัวเลขได้อย่างละเอียด โดยการใช้คอมพิวเตอร์เป็นตัวกำหนดและจัดเก็บตัวเลขเหล่านั้นไว้

สเต็ปเปอร์มอเตอร์สามารถทำงานในระบบเปิด (Open loop system) แต่วิธีการกำหนดตำแหน่งให้ถูกต้องนั้นจำเป็นต้องมีการป้อนกลับไปยังระบบให้รับรู้ โดยทั่วไปจะใช้การสวิตช์ติดตั้งไว้ที่ตำแหน่งที่ต้องการตรวจจับ (Limit switch) เมื่อสเต็ปเปอร์มอเตอร์เริ่มหมุนจนกระทั่งถึงตำแหน่งของสวิตช์ตรวจจับสัญญาณ ก็จะถูกป้อนกลับเข้าสู่ระบบและทราบการทำงานของสเต็ปเปอร์มอเตอร์ได้ตลอดเวลา ซึ่งโดยปกติในวงจรคอนโทรลเลอร์จะมีการกำหนดจุดอ้างอิงไว้ด้วย เพื่อให้เริ่มต้นทำงานและอ้างอิงตำแหน่งได้อย่างถูกต้อง

เพื่อให้เข้าใจมากขึ้นมาดูหลักการทำงานแบบง่าย ๆ ของสเต็ปเปอร์มอเตอร์แบบยูนิโพลาร์ 4 เฟส ตัวโรเตอร์เป็นแม่เหล็กโดยจะเปลี่ยนทิศทางไปตามสนามแม่เหล็ก การจ่ายพลังงานแก่ขดลวดใดขดลวดหนึ่ง โรเตอร์ จะหมุนไป 90 องศา ดังรูปที่ 2.2 (ก) แต่ถ้าจ่ายพลังงานให้พร้อมกันทั้ง 2 ขด โรเตอร์จะหมุนเพียง 45 องศา ดังรูปที่ 2.2 (ข) ซึ่งแบบหลังจะสร้างแรงบิดได้มากว่าแบบแรก สเต็ปเปอร์มอเตอร์จะมีมุมของการเคลื่อนที่แต่ละสเต็ปเป็น 1.8 องศา ดังนั้นที่โรเตอร์จะต้องมีขั้วแม่เหล็ก 50 ขั้ว (90/50 = 1.8) สเต็ปเปอร์มอเตอร์ 4 เฟส ความจริงแล้วเรียกชื่อยังไม่ถูกต้องนัก น่าจะเรียกว่าเป็นแบบ 2 เฟส มากกว่า ถึงแม้ว่าขดลวดจะมี 4 ขดก็ตาม แต่การทำงานของเฟสที่ 3 และ 4 มีค่าเท่ากับเฟสที่ 1 หรือ 2 การที่มีขดลวด 4 ขด ก็เพื่อให้ง่ายต่อการควบคุม เพียงใช้เพาเวอร์ทรานซิสเตอร์เป็นสวิตช์ 4 ตัว

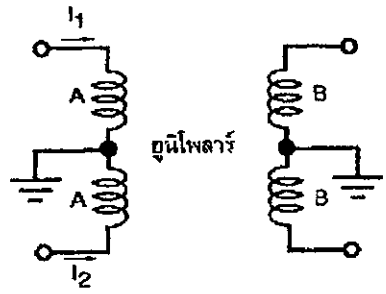


รูปที่ 2.2 ทิศทางการหมุนโรเตอร์ของสเต็ปเปอร์มอเตอร์ 4 เฟส



รูปที่ 2.3 ไบโพลาร์สเต็ปเปอร์มอเตอร์แบบ 2 เฟส
สนามแม่เหล็กจะเปลี่ยนเมื่อกลับทิศทางการไหลของกระแส

ส่วนในรูปที่ 2.3 เป็นการพันขดลวดแบบไบโพลาร์ เมื่อขดลวด A และ B มีกระแสไหลผ่าน สเตเตอร์จะเกิดขั้วแม่เหล็ก เป็นผลให้โรเตอร์ที่มีขั้วแม่เหล็กต่างกับสเตเตอร์ถูกดูด ต่อมาเมื่อกระแสที่ไหลในขดลวด A เปลี่ยนทิศทางกลับ จึงเป็นผลให้ขั้วแม่เหล็กที่แกน A เปลี่ยนจากขั้วใต้เป็นขั้วเหนือ จากขั้วเหนือเป็นขั้วใต้สลับกัน โรเตอร์จึงถูกผลักให้หมุนทวนเข็มไป 90 องศา เครื่องหมายขีด (-) บนอักษร A, B แทนการกลับขั้ว จะสังเกตได้ว่าเมื่อกลับขั้วแม่เหล็กในแต่ละเฟสจะต้องมีการหยุดกระแสก่อนแล้วกระแสจึงค่อยเปลี่ยนทิศทาง จึงสรุปเป็นสเต็ปได้คือ AB, B, AB, A, B, AB, A, AB การทำงานเป็นแบบครึ่งสเต็ปนี้ เป็นผลให้ค่าโมเมนต์มีค่าน้อยกว่าปกติ เพราะมีช่วงเวลาที่กระแสไหลเพียงเฟสเดียว ส่วนแบบยูนิโพลาร์จะคล้ายกับแบบไบโพลาร์โดยคิดขดเดียว ในแต่ละเฟสของยูนิโพลาร์จะมีแทปกลางซึ่งจะแบ่งเป็น 2 ขด ดังรูปที่ 2.4 เป็นผลให้ค่าเส้นแรงแม่เหล็กมีค่าน้อยกว่าไบโพลาร์ ดังนั้นเมื่อสนามแม่เหล็กเปลี่ยนแปลงการะแสจะไม่เปลี่ยนทิศทางการไหล สนามแม่เหล็กที่ได้ก็น้อยตาม แรงบิดที่ขึ้นกับสนามแม่เหล็กก็น้อยกว่าด้วย



รูปที่ 2.4 ยูนิโพลาร์สเต็ปเปอร์มอเตอร์
การเปลี่ยนขั้วแม่เหล็กเกิดจากระแสไหลต่างขดกัน
กระแสจะไม่ไหลพร้อมกัน 2 ขดในสเตเตอร์เดียวกัน

2.1.3 การควบคุมการหมุนของสเต็ปเปอร์มอเตอร์

การกระตุ้นเพื่อควบคุมการหมุนของมอเตอร์ให้เคลื่อนที่ไปแต่ละสเต็ปในการใช้งานจริง ทำได้โดยจ่ายกำลังไฟฟ้าไปยังขดลวดแต่ละขดบนสเตเตอร์ ซึ่งต้องป้อนเป็นแบบซีเควนเชียลในรูปแบบที่ถูกต้อง แบ่งออกได้ 3 รูปแบบ คือ

1.) แบบเวฟ (Wave) เป็นการกระตุ้นรูปแบบที่ง่ายที่สุด โดยกระตุ้นขดลวดทีละขดในเวลาหนึ่งและเรียงถัดกันไป เช่น ขดที่ 1, 2, 3, 4, 1 หรือ 1, 4, 3, 2, 1 ขึ้นอยู่กับทิศทางที่ต้องให้หมุน ดังนั้นจึงมีขดลวดเพียงขดเดียวในเวลาหนึ่งเท่านั้นที่ถูกกระตุ้น วงจรกระตุ้นแบบเวฟมีขั้นตอนการทำงานต่างๆ ดังแสดงในตารางที่2.1

ตารางที่ 2.1 แสดงขั้นตอนการกระตุ้นขดลวดแต่ละเฟสแบบเวฟ

สเต็ป	เฟสที่ 1	เฟสที่ 2	เฟสที่ 3	เฟสที่ 4
1	ทำงาน	-	-	-
2	-	ทำงาน	-	-
3	-	-	ทำงาน	-
4	-	-	-	ทำงาน

2.) แบบ 2 เฟส (Two Phase) เป็นการกระตุ้นอีกรูปแบบหนึ่งซึ่งคล้ายกับแบบเวฟ แต่จ่ายกำลังไฟฟ้าไปที่ขดลวด 2 ขดที่อยู่ใกล้กันในเวลาเดียวกัน และเรียงถัดกันไปเช่นเดียวกับแบบเวฟคือ 12, 23, 34, 41, 12 หรือ 14, 43, 32, 21, 14 ขึ้นอยู่กับทิศทางการหมุน การเพิ่มจำนวนของขดลวดที่ถูกกระตุ้นนี้ทำให้เพิ่มแรงบิดได้มากกว่าแบบเวฟ โรเตอร์จะเคลื่อนที่ด้วยแรงดึงอย่างเต็มแรงจาก 2 ขดที่ถูกกระตุ้น

พร้อมกัน และต่อด้วยแรงดึงจากอีก 2 ขดถัดไป สำหรับข้อเสียของการกระตุ้นแบบนี้ ต้องใช้แหล่งจ่ายกำลังไฟฟ้ามากขึ้น ขั้นตอนการทำงานต่าง ๆ แสดงดังในตารางที่ 2.2

ตารางที่ 2.2 แสดงขั้นตอนการกระตุ้นขดลวดแต่ละเฟสแบบ 2 เฟส

สเต็ป	เฟสที่ 1	เฟสที่ 2	เฟสที่ 3	เฟสที่ 4
1	ทำงาน	ทำงาน	-	-
2	-	ทำงาน	ทำงาน	-
3	-	-	ทำงาน	ทำงาน
4	ทำงาน	-	-	ทำงาน

3.) แบบครึ่งสเต็ป (Half Step) เป็นรูปแบบผสมผสานระหว่างการกระตุ้นแบบเวฟและแบบ 2 เฟส เพื่อเพิ่มจำนวนของสเต็ปต่อรอบอีกหนึ่งเท่าตัว จะกระตุ้นขดลวดเรียงกันไปเป็นลำดับดังนี้ ขดลวด 1, 12, 2, 23, 3, 34, 4, 41, 1 หรือในการหมุนอีกทิศทางหนึ่งจะได้เป็น 1, 14, 4, 43, 3, 32, 2, 21, 1 แรงบิดที่ได้จากการกระตุ้นแบบนี้จะเพิ่มมากขึ้นอีก เพราะช่วงสเต็ปมีระยะสั้นลงและแต่ละสเต็ปเกิดแรงดึงจากขดลวด 2 ขดที่ถูกกระตุ้นพร้อมกัน ความถูกต้องของตำแหน่งมีเพิ่มมากขึ้น ข้อสำคัญคือเมื่อกระตุ้นให้ทำงานในรูปแบบนี้จะต้องทำการหมุนถึง 2 สเต็ปจึงจะได้เท่ากับ 1 สเต็ปเต็มเหมือนการควบคุม 2 แบบแรก สำหรับแหล่งจ่ายกำลังไฟฟ้าต้องใช้เทียบเท่ากับแบบ 2 เฟส จึงจะเพียงพอ ขั้นตอนการทำงานต่าง ๆ แสดงดังในตารางที่ 2.3

ตารางที่ 2.3 แสดงขั้นตอนการกระตุ้นขดลวดแต่ละเฟสแบบครึ่งสเต็ป

สเต็ป	เฟสที่ 1	เฟสที่ 2	เฟสที่ 3	เฟสที่ 4
1	ทำงาน	-	-	-
2	ทำงาน	ทำงาน	-	-
3	-	ทำงาน	-	-
4	-	ทำงาน	ทำงาน	-
5	-	-	ทำงาน	-
6	-	-	ทำงาน	ทำงาน
7	-	-	-	ทำงาน
8	ทำงาน	-	-	ทำงาน

2.2 ส่วนการเขียนโปรแกรม

2.2.1 พื้นฐานการเขียนโปรแกรมด้านฐานข้อมูล [1]

ระบบฐานข้อมูล (Database)

ระบบฐานข้อมูล(Database) คือ ระบบจัดการเก็บข้อมูลที่รวบรวมเอาข้อมูลที่มีความสัมพันธ์กันมาเก็บไว้ด้วยกัน เพื่อลดความซ้ำซ้อนของข้อมูล นอกจากนี้ยังคำนึงถึงการใช้ข้อมูลร่วมกันระหว่างงานหลาย ๆ งาน และความเป็นอิสระต่อกันระหว่างข้อมูลกับงานที่เรียกใช้อีกด้วย

ชนิดของฐานข้อมูล

ระบบฐานข้อมูลได้ถูกพัฒนาให้มีประสิทธิภาพในการจัดเก็บและเรียกใช้โดยจะมีโครงสร้างของการออกแบบฐานข้อมูลแบ่งได้เป็น 3 แบบด้วยกัน ดังนี้

1. ฐานข้อมูลแบบเชิงชั้น (Hierarchical Database) เป็นฐานข้อมูลที่มีการจัดลำดับความสำคัญระหว่างแฟ้มข้อมูล
2. ฐานข้อมูลแบบเครือข่าย (Network Database) เป็นฐานข้อมูลที่มีความสัมพันธ์ระหว่างแฟ้มข้อมูลเชื่อมโยงถึงกันหมด
3. ฐานข้อมูลแบบสัมพันธ์ (Relational Database) เป็นฐานข้อมูลที่จัดเก็บในรูปของตารางข้อมูล โดยข้อมูลแต่ละส่วนจะมีความสัมพันธ์ซึ่งกันและกันนั่นหมายความว่า เมื่อเราทราบข้อมูลส่วนหนึ่งจนจะมีผลให้เราสามารถทราบข้อมูลส่วนอื่นๆ ที่สัมพันธ์กันได้

ความสัมพันธ์ของข้อมูลระหว่างตาราง สามารถแยกออกได้ 3 ชนิด คือ

- One-to-One หมายถึง ข้อมูล 1 เร็ดคอร์ดที่อยู่ในตารางหนึ่ง มีความสัมพันธ์หรือเชื่อมโยงกับอีกตารางหนึ่ง โดยที่สามารถอ้างอิงได้เพียง 1 เร็ดคอร์ดเท่านั้น เช่น นักศึกษา 1 คน มีรหัสประจำตัวได้ 1 หมายเลข ในทางกลับกันรหัสประจำตัว 1 หมายเลข สามารถอ้างถึงนักศึกษาได้เพียง 1 คนเท่านั้น
- One-to-Meny หมายถึง ข้อมูล 1 เร็ดคอร์ดที่อยู่ในตารางหนึ่ง มีความสัมพันธ์หรือเชื่อมโยงกับอีกตารางหนึ่ง โดยที่สามารถอ้างอิงได้มากกว่า 1 เร็ดคอร์ด เช่น มหาวิทยาลัยมีนักศึกษาได้หลายคน ในทางกลับกันมีนักศึกษาหลายคนศึกษาอยู่ในมหาวิทยาลัยเดียวกัน
- Many-to-Many หมายถึง ข้อมูลหลายเร็ดคอร์ดที่อยู่ในตารางหนึ่ง มีความสัมพันธ์หรือเชื่อมโยงกับอีกตารางหนึ่ง โดยที่สามารถอ้างอิงได้มากกว่า 1 เร็ดคอร์ด เช่น นักศึกษา 1 คน ลงทะเบียนได้หลายวิชา และแต่ละวิชามีนักศึกษาลงทะเบียนได้หลายคนเช่นกัน

2.2.2 ระบบฐานข้อมูลของ Microsoft Access [3]

ในระบบ Access จะใช้ระบบฐานข้อมูลแบบสัมพันธ์ (Relational Database) โดยอาศัยรูปแบบของตาราง (Table) เป็นตัวสร้างความสัมพันธ์ระหว่างข้อมูล โดยที่ถ้ามองข้อมูลในลักษณะแนวนอน (Row) จะถูกเรียกว่า เร็ดคอร์ด (Record) และถ้ามองในลักษณะแนวตั้ง (Column) จะถูกเรียกว่า ฟิลด์ (Field) หรือแอ็ตทริบิวต์ (Attribute)

2.2.3 รูปแบบการติดต่อกับฐานข้อมูลด้วย Visual Basic [2]

ในการติดต่อกับฐานข้อมูล โดยปกติแล้ว Visual Basic จะเชื่อมโยงผ่านทาง Database Engine ที่เรียกว่า JET Engine จึงอาจกล่าวได้อีกนัยหนึ่งว่า JET Engine คือ ไดรเวอร์ชนิดหนึ่ง ซึ่งทำหน้าที่เป็นตัวเชื่อมโยงให้ VB สามารถติดต่อกับฐานข้อมูลได้นั่นเอง โดยที่ฐานข้อมูลหลัก (Default) ที่ VB รู้จักเป็นอย่างดีคือ MS Access แต่ Visual Basic สามารถติดต่อกับฐานข้อมูลได้ทุกชนิดเช่นกัน โดยอาศัยเทคโนโลยีหลายๆ อย่าง

สำหรับฐานข้อมูลจะใช้ฐานข้อมูลของ MS Access 2000 Thai Edition เป็นหลักดังนั้นผู้ใช้สามารถใช้งาน MS Access 2000 ได้บ้าง ก็จะทำให้การสร้างแอพพลิเคชันด้านฐานข้อมูลด้วย Visual Basic การติดต่อฐานข้อมูลใน Visual Basic จะแยกเป็น 4 ประเภทใหญ่ๆ คือ

แบบที่ 1 – ติดต่อโดยอาศัยคอนโทรลด้านฐานข้อมูล

มีศัพท์เรียกคอนโทรลกลุ่มนี้โดยเฉพาะว่า Bound Controls ส่วนใหญ่แล้วก็คือ กลุ่มของคอนโทรลมาตรฐานที่ใช้กันโดยทั่วไปนั่นเอง เช่น คอนโทรล TextBox, PictureBox, Image, ListBox. ComboBox เป็นต้น โดยใช้คอนโทรล Data (Data Control) เป็นตัวเชื่อมระหว่างฐานข้อมูลกับกลุ่ม Bound Controls

ผู้ใช้สามารถตรวจสอบได้ว่า คอนโทรลตัวใดบ้างถูกจัดอยู่ในกลุ่มของ Bound Controls โดยการตรวจสอบว่าคอนโทรลตัวนั้น มีคุณสมบัติที่ขึ้นต้นด้วยคำว่า Data... เช่น DataField, DataFormat, DataSource หรือไม่ ถ้ามีหมายถึง คอนโทรลตัวดังกล่าว ถูกจัดอยู่ในกลุ่มของ Bound Controls ด้วยเช่นกัน

ยังมีคอนโทรลอีกกลุ่มที่เรียกว่า ActiveX Bound Controls หมายถึง กลุ่มของคอนโทรลที่มีคุณสมบัติที่ขึ้นต้นด้วยคำว่า Data... เช่นกัน แต่มีข้อแตกต่างจากคอนโทรลในกลุ่ม Bound Controls ก็คือ กลุ่ม ActiveX Bound Controls อาจจะมาจากผู้ผลิตรายอื่นๆ (Thirds Party) ซึ่งเป็นคอนโทรลที่ไม่ได้อยู่ในแถบเครื่องมือมาตรฐานของ Visual Basic โดยที่คุณต้องเพิ่มเติมคอนโทรลกลุ่มนี้เข้ามาในแถบเครื่องมือเอง ดังนั้นจึงเรียกคอนโทรลกลุ่มนี้ว่า ActiveX Bound Controls เช่น คอนโทรล DBGrid, คอนโทรลDBList หรือคอนโทรล DBCombo ฯลฯ เป็นต้น

แบบที่ 2 – ติดต่อโดยใช้ออบเจ็กต์ Data Access Object (DAO)

ถือว่าเป็นวิธีที่ล้าสมัยแล้ว โดยมีแนวคิดในการติดต่อหรือเข้าถึงข้อมูลในฐานข้อมูลผ่านทางองค์ประกอบต่างๆ ในฐานข้อมูล เช่น ฟิลด์ (Field), เร็คคอร์ด (Record), ความสัมพันธ์ระหว่างตาราง (Relation) เป็นต้น โดยจะแทนแต่ละองค์ประกอบเหล่านั้นด้วยออบเจ็กต์ (Object) และควบคุมออบเจ็กต์ต่าง ๆ เหล่านี้โดยการเขียนโค้ด

แม้จะทำงานได้ดีกว่า อิสระกว่า แต่มีความยุ่งยากในการเขียนโปรแกรมด้วยเช่นกัน อีกทั้งเป็นเทคโนโลยีที่เก่าแล้วคือ เน้นเฉพาะระบบฐานข้อมูลที่เป็นตาราง (โดยเฉพาะ Access รุ่นเก่า ๆ) แต่การเก็บข้อมูลในปัจจุบัน ถูกเก็บอยู่ในสภาพแวดล้อมแตกต่างกันมากมาย เช่น รูปภาพ (Image), ข้อความ (Text) และรูปแบบอื่น ๆ อีกมากมายทำให้ต้องสร้างออบเจ็กต์ใหม่ ๆ ขึ้นมาเรื่อยๆ แต่นั่นไม่ใช่สิ่งที่ทำได้ง่าย และกลายเป็นข้อจำกัดที่สำคัญของ DAO ตัวอย่างของออบเจ็กต์ในกลุ่มนี้ เช่น ออบเจ็กต์ RecordSet, ออบเจ็กต์ TableDef, คอลเล็กชัน Fields เป็นต้น

แบบที่ 3 – ติดต่อผ่าน ODBC โดยตรง (ODBC Direct)

เป็นการติดต่อกับฐานข้อมูลแบบ 32 บิตที่สนับสนุนมาตรฐาน ODBC (Open DataBase Connectivity) ที่ JET Engine (กลไกการติดต่อฐานข้อมูลของไมโครซอฟท์ ซึ่งเน้นที่ Access) ไม่สามารถจัดการได้ เช่น ฐานข้อมูลของ Oracle, ฐานข้อมูลของ Microsoft SQL Server เป็นต้น ซึ่งเป็นการติดต่อเฉพาะฐานข้อมูลที่มีการเก็บข้อมูลในรูปแบบตารางเท่านั้น

ไมโครซอฟท์เองได้สร้างออบเจ็กต์ขึ้นมาอีกชุดหนึ่งที่ชื่อว่า Remote Data Object หรือที่เรียกสั้น ๆ ว่า RDO เพื่อใช้สำหรับติดต่อกับฐานข้อมูลแบบเครือข่าย (เช่น Oracle, SQL Server, DB2 ฯลฯ) ซึ่งเป็นไปตามสถาปัตยกรรม n-Tier (n เทียร์) เช่น Client/Server (2-Tier ทูเทียร์) หรือ Middle Tier (3-Tier ทรีเทียร์) โดยอาศัยมาตรฐาน ODBC ในการเชื่อมโยงที่เกี่ยวข้องกับ Visual Basic คือ คอนโทรล Remote Data (Remote Data Control-RDC) และออบเจ็กต์ Remote Data (Remote Data Object - RDO)

แบบที่ 4 – เข้าถึงข้อมูลโดยอาศัยเทคโนโลยี OLEDB

เป็นรูปแบบการติดต่อกับฐานข้อมูลผ่านทางกลุ่มออบเจ็กต์ในโมเดล ADO ซึ่งใช้ OLEDB Provider เป็นกลไกในการเข้าถึงข้อมูลในฐานข้อมูลแทน JET Engine โดยเปลี่ยนจากมุมมองการติดต่อฐานข้อมูลแบบเดิม ๆ ที่ต้องกำหนดชนิดของฐานข้อมูลมาเป็นมองที่รูปแบบของการเชื่อมต่อ (Connection) เข้ากับฐานข้อมูล

2.2.4 การใช้งานภาษา SQL [3]

SQL ย่อมาจากคำว่า Structured Query Language หมายถึง ภาษามาตรฐานกลางที่ใช้สำหรับจัดการข้อมูลในฐานข้อมูล โดยเฉพาะอย่างยิ่งฐานข้อมูลประเภท RDBMS (Relation Database System) จะรู้จักภาษา SQL เป็นอย่างดี เราจะใช้ SQL เพื่อจัดการกับข้อมูลในฐานข้อมูลได้หลายอย่าง เช่น การแสดงข้อมูลจากฐานข้อมูลแบบมีเงื่อนไข, การเพิ่ม, การลบ และการนำข้อมูลจากตารางหลาย ๆ ตารางมาแสดงร่วมกันได้ เป็นต้น เป็นภาษาที่ใช้จัดการข้อมูลในฐานข้อมูลได้มีประสิทธิภาพมากที่สุด

เราจะใช้ภาษา SQL เพื่อทำคิวรี (Query) ข้อมูลที่อยู่ในตาราง ในวัตถุประสงค์ที่ต่าง ๆ กัน เช่น อาจจะต้องการข้อมูลที่มาจากตารางเดียว หรือหลายตารางมาแสดงด้วยกันในเวลาเดียวกัน ดังนั้นการทำคิวรีจึงเป็นการสร้างตารางเสมือนขึ้นมา ประกอบด้วยข้อมูลที่มาจากตารางเดียวหรือหลายตารางก็ได้ เป็นตารางที่ไม่มีอยู่จริงในฐานข้อมูล เป็นมุมมองของข้อมูลในฐานข้อมูลตามที่คุณต้องการ

โครงสร้างของภาษา SQL

ภาษา SQL ประกอบไปด้วย 3 ส่วนใหญ่ๆ คือ

1. Data Definition Language(DDL) – เป็นกลุ่มคำสั่งในภาษา SQL ที่ใช้สำหรับจัดการโครงสร้างของฐานข้อมูล เช่น การสร้างฐานข้อมูล, ปรับปรุงโครงสร้างของฐานข้อมูล เป็นต้น ตัวอย่างการใช้งานกลุ่มคำสั่ง DDL นี้ก็คือ การสร้างฐานข้อมูลด้วย MS SQL Server 7.0 ก็จะมีการใช้งานคำสั่งในกลุ่ม DDL เป็นหลัก
2. Data Manipulation Language(DML) – เป็นกลุ่มคำสั่งในภาษา SQL ที่ใช้สำหรับจัดการข้อมูลในฐานข้อมูล เช่น การแสดงข้อมูลแบบมีเงื่อนไข , การลบข้อมูล, การเพิ่มข้อมูล และการแสดงข้อมูลที่มาจากตารางหลายตาราง เป็นต้น
3. กลุ่มฟังก์ชัน Aggregate Function – เป็นฟังก์ชันพิเศษของภาษา SQL ที่ทำหน้าที่เฉพาะอย่าง เช่น หาผลรวมเร็ดคอร์ด, ค่าสูงสุด , ค่าต่ำสุด เป็นต้น เป็นกลุ่มฟังก์ชันที่มีประโยชน์มาก เพราะจะช่วยลดภาระให้คุณไม่ต้องเขียนโค้ดจัดการเอง

สำหรับการใช้งานภาษา SQL ร่วมกับ Visual Basic เพื่อจัดการข้อมูลในฐานข้อมูล จะใช้งานกลุ่มคำสั่ง DML เป็นหลัก ร่วมกับกลุ่มฟังก์ชัน Aggregate และกำหนดเงื่อนไขโดยการใช้ตัวดำเนินการด้านต่าง ๆ

2.3 การ์ด ET-PC 8255

ET-PC 8255 จะเป็นการ์ดต่อระบบขยายของเครื่อง PC ให้มีส่วนของพอร์ทอินพุท-เอาท์พุทที่สามารถใช้งานได้มากขึ้นโดยจะมีพอร์ทที่ใช้งานเป็นอินพุท-เอาท์พุทจำนวน 9 พอร์ทหรือ 72 บิท

2.3.1 การทำงานของ ET-PC 8255

การ์ด ET-PC 8255 จะประกอบไปด้วยสองส่วนใหญ่ๆ คือ ส่วนไอซี 8255 ซึ่งทำหน้าที่เป็นพอร์ทอินพุท-เอาท์พุท และส่วนของวงจรไอซี decode (เลือกตำแหน่งของพอร์ท) คือไอซีหมายเลข 74LS688 และ 74LS139

2.3.2 การใช้งาน ไอซี 8255

ไอซี 8255 ประกอบไปด้วยพอร์ทการทำงาน 3 พอร์ท และอีกหนึ่งพอร์ทที่ใช้ในการควบคุม ก่อนที่เราจะใช้งาน ไอซี 8255 เราจะต้องส่งข้อมูลไปยังพอร์ทควบคุมก่อนว่าจะให้ทั้งสามพอร์ทของ I ไอซี 8255 ทำหน้าที่เป็นพอร์ทอินพุทหรือเอาท์พุท

ขาต่างๆของไอซี 8255

- **D0-D7** เป็นขาข้อมูลอินพุทและเอาท์พุทที่จะต้องผ่านเข้าออกจากส่วนนี้ ดังนั้น D0-D7 จึงเป็นส่วนที่จะต่อเข้ากับระบบบัสของคอมพิวเตอร์เพื่อให้คอมพิวเตอร์สามารถอ่านหรือเขียนข้อมูลจากพอร์ทผ่านทางบัสนี้
- **CS** เป็นขาอินพุทที่จะรับค่ามาภายนอกเพื่อเลือกชิพ 8255 โดยเมื่อขานี้เป็นลอจิก 0 จะทำให้ 8255 ต่อเข้ากับระบบบัสของคอมพิวเตอร์เพื่อให้คอมพิวเตอร์เขียนหรืออ่านข้อมูลจากพอร์ทได้
- **RD** เป็นขาสัญญาณการอ่าน เป็นสัญญาณอินพุทที่ส่งมาจาก CPU เมื่อสัญญาณนี้เป็นลอจิก 0 และ CS เป็นลอจิก 0 ด้วยไอซี 8255 จะทำให้ตัว CPU อ่านข้อมูลจากบัสในขณะที่เป็นพอร์ทอินพุท
- **WR** เป็นขาสัญญาณการเขียนจะทำงานเมื่อสัญญาณ WR เป็นลอจิก 0 และ CS เป็นลอจิก 0 สัญญาณนี้มาจาก CPU เมื่อต้องการเขียนข้อมูลลงพอร์ทที่กำหนด
- **A0 – A1** เป็นขา Address ลอจิกของทั้ง 2 ขานี้จะถอดรหัสออกมาได้ 4 ค่า เพื่อกำหนดค่า Register ภายในที่เชื่อมต่อเข้ากับพอร์ทอินพุทและเอาท์พุทของไอซี 8255 เพื่อเลือกใช้พอร์ท A,B และ C
- **RESET** เป็นขารีเซต เป็นขาสัญญาณที่ส่งมาจากภายนอกเพื่อทำการเคลียสถานะของ ไอซี 8255 เมื่อ ไอซี 8255 ได้รับการรีเซตมันจะกลับเข้าสู่โหมดอินพุท หรือทุกพอร์ทเป็นพอร์ทอินพุท
- **PA0 – PA7** เป็นสายสัญญาณที่เป็นพอร์ทของไอซี 8255 ชื่อพอร์ท A การเลือกพอร์ทจะเลือกโดยขา A0 – A1
- **PB0 – PB7** เป็นสายสัญญาณที่เป็นพอร์ทของไอซี 8255 ชื่อพอร์ท B การเลือกพอร์ทจะเลือกโดยขา A0 –A1
- **PC0 – PC7** เป็นสายสัญญาณที่เป็นพอร์ทของไอซี 8255 การกำหนดพอร์ทนี้จะได้รับการกำหนดโดยขาแอดเดรส A0 - A1 พอร์ท C นี้แบ่งออกได้ 2 กลุ่ม คือ PC0 – PC3 และกลุ่ม PC4 –PC7